

Aufgabe 1

Die Syntax von WHILE sei um die Regel

1 $C ::= \text{repeat } C \text{ until } B$

erweitert. Ergänzen Sie die operationelle Semantik von WHILE, so dass diese zusätzliche Anweisungsform angemessen behandelt wird.

$$\Delta \langle W | S | \text{repeat } C \text{ until } B . K | E | A \rangle = \langle W | S | C . \text{while } B \text{ do } C . K | E | A \rangle$$

C wird mindestens einmal ausgeführt, anschliessend verhält sich die „repeat C until B“-Schleife identisch zu „while B do C“.

Aufgabe 2

Erweitern Sie die Syntax von WHILE, so dass in den booleschen Ausdrücken auch die booleschen Operatoren *and* und *or* vorkommen dürfen. Geben Sie für diese Erweiterung eine operationelle Semantik an, die eine nicht-strikte Semantik von *and* und *or* festlegt.

$$B ::= W | \text{not } B | T_1 \text{ BOP } T_2 | B_1 \text{ and } B_2 | B_1 \text{ or } B_2 | \text{read}$$

And:

$$\Delta \langle W | S | B_1 \text{ and } B_2 . K | E | A \rangle = \langle W | S | B_1 . \text{and} . B_2 . K | E | A \rangle$$

$$\Delta \langle \text{true} . W | S | \text{and} . B_2 . K | E | A \rangle = \langle W | S | B_2 . K | E | A \rangle$$

$$\Delta \langle \text{false} . W | S | \text{and} . B_2 . K | E | A \rangle = \langle \text{false} . W | S | K | E | A \rangle$$

Or:

$$\Delta \langle W | S | B_1 \text{ or } B_2 . K | E | A \rangle = \langle W | S | B_1 . \text{or} . B_2 . K | E | A \rangle$$

$$\Delta \langle \text{false} . W | S | \text{or} . B_2 . K | E | A \rangle = \langle W | S | B_2 . K | E | A \rangle$$

$$\Delta \langle \text{true} . W | S | \text{or} . B_2 . K | E | A \rangle = \langle \text{true} . W | S | K | E | A \rangle$$

Aufgabe 3

Erweitern Sie die WSKEA-Maschine um eine Komponente *N* für Nachrichten (Texte), in der kurze, sinnvolle Meldungen eingetragen werden, wenn es keinen Folgezustand gibt, oder wenn die Ausführung korrekt terminiert.

Grundsätzlich identisch zur WSKEA-Maschine

Grundzustand der WSKEA-Maschine ist: $\langle W | S | K | E | A | \epsilon \rangle$

Falls das Programm korrekt terminiert gilt: $\Delta \langle W | S | \epsilon | E | A | \epsilon \rangle = \langle W | S | \epsilon | E | A | \text{„Terminiert“} \rangle$

In einem Fehlerzustand gibt die Maschine eine informative Fehlermeldung zurück. Ein Beispiel wäre:

$$\Delta \langle 0.n . W | S | / . K | E | A | \epsilon \rangle = \langle W | S | K | E | A | \text{„Fehler: Division durch 0“} \rangle$$

Die Darstellung weiterer Fehlerzustände ist analog dazu.

Aufgabe 4 (freiwillig)

Implementieren Sie in einer Sprache Ihrer Wahl

- den Zustandsraum der WSKEA-Maschine,
- eine Funktion `anfang`, die zu einem WHILE-Programm und einer Eingabe den Anfangszustand ergibt, und
- die Zustandsüberföhrungsfunktion `delta`.

Lösung in **Whitespace** ;-)